

We claim:

1 1. A method of upgrading a software application from a user modified prior
2 version to an upgrade version, wherein the user modified prior version and the upgrade
3 version have a common ancestor version, said method comprising:

4 determining a first set of differences based on a comparison of the user
5 modified prior version and the common ancestor version;

6 determining a second set of differences based on a comparison of the upgrade
7 version and the common ancestor version;

8 determining which differences from said first and second sets of differences are
9 compatible differences and which are conflicting differences; and

10 applying changes to the upgrade version associated with said compatible
11 differences.

1 2. The method of claim 1, wherein the user modified prior version comprises a
2 first plurality of objects and the common ancestor version comprises a second plurality
3 of objects, and wherein said determining said first set of differences comprises:

4 determining whether one or more objects from said first plurality of objects
5 share a common name with one or more objects from said second plurality of objects,
6 and if so, determining whether attributes associated with the commonly named objects
7 are identical, and if not, including data related to the difference between said attributes
8 in said first set of differences.

1 3. The method of claim 2, wherein the difference between said attributes is
2 determined to be a compatible difference if said attributes are superficial.

1 4. The method of claim 3, wherein said attributes are designated as being
2 superficial by a user.

1 5. The method of claim 2, wherein the upgrade version comprises a third plurality
2 of objects, and wherein said determining said second set of differences comprises:

3 determining whether one or more objects from said second plurality of objects
4 share a common name with one or more objects from said third plurality of objects, and

5 if so, determining whether attributes associated with the commonly named objects are
6 identical, and if not, including data related to the difference between said attributes in
7 said second set of differences.

1 6. The method of claim 1, further comprising determining a third set of differences
2 based on a comparison of the upgrade version and the user modified prior version.

1 7. The method of claim 6, wherein the user modified prior version comprises a
2 first plurality of objects, the common ancestor version comprises a second plurality of
3 objects, and the upgrade version comprises a third plurality of objects, and wherein said
4 determining said third set of differences comprises:

5 determining whether an object from said first plurality of objects is not included
6 within said second or third plurality of objects, and if so, indicating that the addition of
7 said object to said first plurality of objects is a compatible difference.

1 8. The method of claim 7, wherein said applying changes to the upgrade version
2 associated with said compatible differences comprises adding said object to the upgrade
3 version.

1 9. The method of claim 6, wherein the user modified prior version comprises a
2 first plurality of objects, the common ancestor version comprises a second plurality of
3 objects, and the upgrade version comprises a third plurality of objects, and wherein said
4 determining said third set of differences comprises:

5 determining whether an object from said second and third plurality of objects is
6 not included within said first plurality of objects, and if so, indicating that the deletion
7 of said object from said first plurality of objects is a conflicting difference.

1 10. The method of claim 9, wherein said applying changes to the upgrade version
2 associated with said compatible differences comprises adding said object to the upgrade
3 version.

1 11. Computer executable software code for upgrading a software application from a
2 user modified prior version to an upgrade version, wherein the user modified prior

3 version and the upgrade version have a common ancestor version, said software code
4 comprising:

5 code to determine a first set of differences based on a comparison of the user
6 modified prior version and the common ancestor version;

7 code to determine a second set of differences based on a comparison of the
8 upgrade version and the common ancestor version;

9 code to determine which differences from said first and second sets of
10 differences are compatible differences and which are conflicting differences; and

11 code to apply changes to the upgrade version associated with said compatible
12 differences.

1 12. The software code of claim 11, wherein the user modified prior version
2 comprises a first plurality of objects and the common ancestor version comprises a
3 second plurality of objects, and wherein said code to determine said first set of
4 differences comprises:

5 code to determine whether one or more objects from said first plurality of
6 objects share a common name with one or more objects from said second plurality of
7 objects, and if so, to determine whether attributes associated with the commonly named
8 objects are identical, and if not, to include data related to the difference between said
9 attributes in said first set of differences.

1 13. The software code of claim 12, wherein the difference between said attributes is
2 determined to be a compatible difference if said attributes are superficial.

1 14. The software code of claim 12, wherein the upgrade version comprises a third
2 plurality of objects, and wherein said code to determine said second set of differences
3 comprises:

4 code to determine whether one or more objects from said second plurality of
5 objects share a common name with one or more objects from said third plurality of
6 objects, and if so, to determine whether attributes associated with the commonly named
7 objects are identical, and if not, to include data related to the difference between said
8 attributes in said second set of differences.

1 15. The software code of claim 11, further comprising code to determine a third set
2 of differences based on a comparison of the upgrade version and the user modified
3 prior version.

1 16. The software code of claim 15, wherein the user modified prior version
2 comprises a first plurality of objects, the common ancestor version comprises a second
3 plurality of objects, and the upgrade version comprises a third plurality of objects, and
4 wherein said code to determine said third set of differences comprises:

5 code to determine whether an object from said first plurality of objects is not
6 included within said second or third plurality of objects, and if so, to indicate that the
7 addition of said object to said first plurality of objects is a compatible difference.

1 17. The software code of claim 16, wherein said code to apply changes to the
2 upgrade version associated with said compatible differences comprises code to add said
3 object to the upgrade version.

1 18. The software code of claim 17, wherein the user modified prior version
2 comprises a first plurality of objects, the common ancestor version comprises a second
3 plurality of objects, and the upgrade version comprises a third plurality of objects, and
4 wherein said code to determine said third set of differences comprises:

5 code to determine whether an object from said second and third plurality of
6 objects is not included within said first plurality of objects, and if so, to indicate that the
7 deletion of said object from said first plurality of objects is a conflicting difference.

1 19. The software code of claim 18, wherein said code to apply changes to the
2 upgrade version associated with said compatible differences comprises code to add said
3 object to the upgrade version.

1 20. An apparatus for upgrading a software application from a user modified prior
2 version to an upgrade version, wherein the user modified prior version and the upgrade
3 version have a common ancestor version, said apparatus comprising:

4 means for determining a first set of differences based on a comparison of the
5 user modified prior version and the common ancestor version;

6 means for determining a second set of differences based on a comparison of the
7 upgrade version and the common ancestor version;

8 means for determining which differences from said first and second sets of
9 differences are compatible differences and which are conflicting differences; and

10 means for applying changes to the upgrade version associated with said
11 compatible differences.

1 21. The apparatus of claim 20, wherein the user modified prior version comprises a
2 first plurality of objects and the common ancestor version comprises a second plurality
3 of objects, and wherein said means for determining said first set of differences
4 comprises:

5 means for determining whether one or more objects from said first plurality of
6 objects share a common name with one or more objects from said second plurality of
7 objects, and if so, for determining whether attributes associated with the commonly
8 named objects are identical, and if not, for including data related to the difference
9 between said attributes in said first set of differences.

1 22. A method of upgrading a software application from a user modified prior
2 version to an upgrade version, wherein the user modified prior version and the upgrade
3 version have a common ancestor version, said method comprising:

4 comparing the user modified prior version, the common ancestor version, and
5 the upgrade version to determine differences;

6 determining which of said differences are compatible and which are conflicting;
7 and

8 applying changes to the upgrade version associated with said compatible
9 differences.

1 23. The method of claim 22, wherein the user modified prior version comprises a
2 first plurality of objects, the common ancestor version comprises a second plurality of
3 objects, the upgrade version comprises a third plurality of objects, and wherein said
4 comparing comprises comparing said first plurality of objects with said second and
5 third pluralities of objects to determine whether objects were added, deleted, or
6 modified by a user.

1 24. The method of claim 23, wherein said determining comprises:
2 indicating that differences associated with objects added by said user are
3 compatible differences;
4 indicating that differences associated with objects deleted by said user are
5 conflicting differences; and
6 indicating that differences associated with objects modified by said user are
7 compatible differences if said objects modified by said user are superficial.

1 25. Computer executable software code for upgrading a software application from a
2 user modified prior version to an upgrade version, wherein the user modified prior
3 version and the upgrade version have a common ancestor version, said software code
4 comprising:
5 code to compare the user modified prior version, the common ancestor version,
6 and the upgrade version to determine differences;
7 code to determine which of said differences are compatible and which are
8 conflicting; and
9 code to apply changes to the upgrade version associated with said compatible
10 differences.

1 26. The software code of claim 25, wherein the user modified prior version
2 comprises a first plurality of objects, the common ancestor version comprises a second
3 plurality of objects, the upgrade version comprises a third plurality of objects, and
4 wherein said code to compare comprises code to compare said first plurality of objects
5 with said second and third pluralities of objects to determine whether objects were
6 added, deleted, or modified by a user.

1 27. The software code of claim 26, wherein said code to determine comprises:
2 code to indicate that differences associated with objects added by said user are
3 compatible differences;
4 code to indicate that differences associated with objects deleted by said user are
5 conflicting differences; and

6 code to indicate that differences associated with objects modified by said user
7 are compatible differences if said objects modified by said user are superficial.

ALL INFORMATION CONTAINED
HEREIN IS UNCLASSIFIED
DATE 10/22/2018 BY SP/SP